

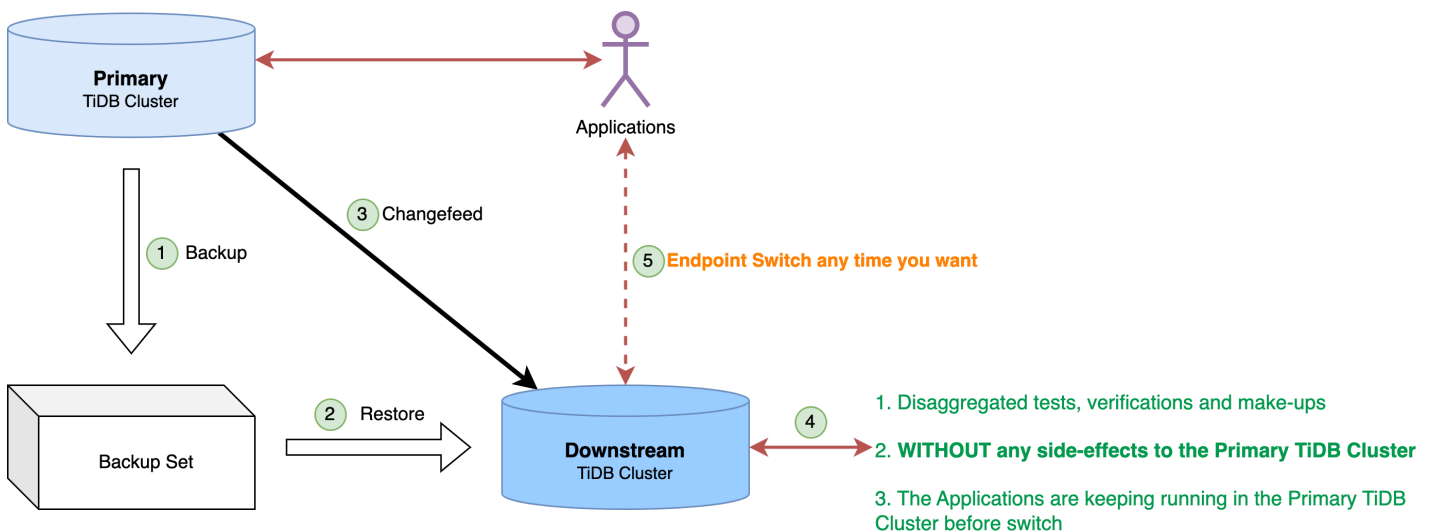
Cluster Migration: User Manual of Backup/Restore/Changefeed

1. Brief Introduction

Almost each TiDB Cluster already has `auto-backup` and `Backup Set` is generated in the backend. Then, we can **create a new TiDB Cluster** and **restore all the data** from the `Backup Set` of the primary TiDB Cluster.

Also, with the capability of `TiCDC`, we can replicate the incremental data change to the restored TiDB Cluster continuously.

As such, we can migrate the primary TiDB Cluster from old architecture to the new **with a much lower risk, and which has very little side-effects to the primary**.



As picture shows above:

1. The TiDB Cluster has been backup automatically. Or backup the TiDB Cluster manually with very little resource consuming.
2. On the WebConsole, the user can simply **restore the `Backup Set` to a new TiDB Cluster with only several clicks.**

3. On the WebConsole, the user can simply **create a Changefeed on the primary with only several clicks**, *which will replicate all incremental data changes to the new restored TiDB Cluster*.
4. Then, we can **do all kinds of tests and verifications on the new TiDB Cluster, *without any side-effects to the Primary TiDB Cluster***. Which will **highly reduce all kinds of risks**.
5. After the tests and verifications, the customer can switch the DB address of the Applications any time you want.

Such a solution has advantages:

1. Much lower risk, all operations and tests run on a distinguished cluster which has no side-effects to the primary.
2. Almost zero cost.
3. Simple clicks, easy to operate.
4. Anytime the customer can choose to do the *Application Switch*.

2. Detailed Steps

Whole stages are:

1. Preparation before starting, for example, index validation, stop the GC, etc.
2. Backup the TiDB cluster
3. Restore the backup set to a new cluster
4. Create Changefeed
5. After synchronization, Verification and Make-up
 - a. If OK, continue to next.
 - b. If not OK, just delete the new TiDB Cluster. Which will have nothing related to the primary one.
6. Application Switch
 - a. Application test on the new TiDB cluster
 - b. Switch the DB address in Application

2.1 Before Start

Before everything starts, the user should take care of the following issues:

- **Cluster Assessment**, to estimate how long it may take to do the backup and restore, and how long it will reach synchronized after Changefeed is created.
 - How many writes per second
 - How many schema objects
 - Storage size
 - Whether going to do DDL or not, before Application Switch
- **Keep GC Safepoint static**. We have to keep the `tikv_gc_safe_point` of the primary cluster to **be static** unless the Changefeed is created. In two ways (just choose one):
 - **Set `tidb_gc_life_time`**, the value should be more than the total sum time below:
 - Backup time, which depends on the total storage, db/table count, region count and etc. If the total storage is less than 500 GB, then you don't need to bother.
 - Restore time, around 1.5TB per hour.
 - Operation time to create Changefeed, which should be soon.
 - And all the time between each operation.

```
1 // We'd better give enough time windows for the Changefeed to
2 // be created. The total time depends how long it take to do
3 // the backup, restore and creation of Changefeed, as well
4 // as the gap time between each operation.
5 // The value '168h' below is about 7 days which should be
6 // long enough.
7 //
8 // In earlier versions 6.1.x, the value can only be up to '24h',
9 // which should be enough as well if you do it continually.
10 set global tidb_gc_life_time=168h;
```

- **Set `tidb_gc_enable`**, then the GC will be disabled. We'd have to enable it after the Changefeed is created.

```
1 set global tidb_gc_enable=off;
```

- **Side-effect if GC is disabled or longer `tidb_gc_life_time`**
 - All delta data will not be compacted, which may **take more storage** than expected.

- The volume it takes depends on **how much write-workload** the Application has.
- **User should confirm that the storage of the TiDB cluster should be enough for the coming delta data size in '168h' (or to the time the GC is enable).**

```
mysql> show variables like '%gc%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| tidb_enable_gc_aware_memory_track | OFF |
| tidb_enable_gogc_tuner | ON |
| tidb_gc_concurrency | -1 |
| tidb_gc_enable | ON |
| tidb_gc_life_time | 10m0s |
| tidb_gc_max_wait_time | 86400 |
| tidb_gc_run_interval | 10m0s |
| tidb_gc_scan_lock_mode | LEGACY |
| tidb_gogc_tuner_threshold | 0.6 |
| tidb_server_memory_limit_gc_trigger | 0.7 |
+-----+-----+
10 rows in set (0.26 sec)

mysql> set global tidb_gc_life_time=168h;
Query OK, 0 rows affected (0.34 sec)

mysql> show variables like '%tidb_gc_life_time%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| tidb_gc_life_time | 168h0m0s |
+-----+-----+
1 row in set (0.26 sec)
```

- Make sure that all tables have the UNIQUE KEY or Primary Key, refer to [TiCDC FAQs](#).
- Check the capacity of the CIDR address left, which should be enough for TiDB deployment of two TiDB clusters.
- Take care of the limits of Backup, Restore and Changefeed.

2.2 Backup

Set auto-backup of the TiDB Cluster, to let it do the backup at the time we expected automatically.

Or, backup the TiDB Cluster manually.

2.2.1 Before Manual Backup

The User should take care of the following issues before backup:

- How long it may take to finish the backup.
 - Data volume. If the total storage size is **less than 500 GB, then it should not be a problem.**
 - Database count, table count, index count and etc. It will take more time if there are a lot of tables, even though the total storage size is not that big.
- The Backup operation may take some resources (typically is low), the User should choose the time window with low workload.

2.2.2 Manual Backup

Or the User could just manually backup the TiDB cluster at any time:

- Click the 'Cluster Name' you need to migrate.

Clusters

Cluster Name	Tier Type	Status	Cloud	Region	Version
Cluster0	Dedicated	● Available	AWS	Oregon (us-west-2)	v7.1.2

- Choose the 'Backup' tab on the left.

Cluster0

Overview

Chat2Query

Data

Import

Changefeed

Backup

Overview

Cluster Properties

Cluster Name	Cluster0
Status	● Available
Tier Type	Dedicated Tier
TiDB Version	v7.1.2
Cloud Provider	AWS
Region	Oregon (us-west-2)

- Click the '...' on the right of 'Backup Setting', then you will see the 'Manual Backup' and click it.

Cluster0	Backup							Backup Setting	...
	Overview								
	Chat2Query								
Data								Manual Backup	
Import								Restore	
Changefeed								Restore History	
Backup									
		Name	Backup Time	Type	TSO	Size	Region	Status	
		ab-1373933076662920594-2023-10-31t10-00-09	2023-10-31 10:00:47 UTC±00:00	Auto	445316188602368000	354.8 KiB	us-west-2	Success	...
		verification	2023-10-31 08:46:22 UTC±00:00	Manual	445315018129408000	354.8 KiB	us-west-2	Success	...

- Input the name you want, and click the 'confirm'.

Manual Backup

Name

firstBackupSet

4~16 characters that can only include numbers, lowercase or uppercase letters, and hyphens. The first and last character must be a letter or number.

Start Now 2023-10-31 13:36:36 UTC±00:00

Cancel

Confirm

- Check whether it is finished successfully, and remember the TSO which will be used to create Changefeed later on.

Backup							Backup Setting	...
Name	Backup Time	Type	TSO	Size	Region	Status	Action	
firstBackupSet	2023-11-01 02:58:12 UTC±00:00	Manual	445332191182848000	349.2 KiB	us-west-2	Success	...	

2.3 Restore

2.3.1 Before Restore

Before we do the restore, the User should take care of the following issues:

- Only in the same Project.
- It should be in the same region.
- *Take great care of the CIDR and the left IP count that can be used.*
- Take great care of the data that can not be restored:
 - `mysql.stat_*`
 - `mysql.tidb`
 - `mysql.global_variables`
 - `mysql.sqlbindings`
 - ~~`mysql.capture_plan_baselines_blacklist`, no care~~
 - ~~`mysql.column_stats_usage`, no care~~
 - `mysql.gc_delete_range`
 - `mysql.gc_delete_range_done`
 - `mysql.schema_index_usage`
- The TiDB version of the new cluster should be the same.
- The resource specification should be the same.
- Take care of which backup set is used.
- Take care of the TS of the backup set.
- The restore speed of a single TiKV node can reach 100 MiB/s.

2.3.2 Restore Steps

The steps are:

- Double check the GC configuration of the primary cluster, which should be disabled or set with a big value of `tidb_gc_life_time`, such as '168h', or '24h' in 6.1.x.

```
1 show variables like '%gc%';
```

- Choose the backup set you want to restore.

Backup							
Backup Setting							
Name	Backup Time	Type	TSO	Size	Region	Status	Action
secondBackupSet	2023-11-01 06:23:24 UTC±00:00	Manual	445335418699776000	357.0 KiB	us-west-2	Success	...
firstBackupSet	2023-11-01 02:58:12 UTC±00:00	Manual	445332191182848000	349.2 KiB	us-west-2	Success	Restore Delete

- Choose the backup set, the region and specify the name.

1

Restore Mode & Restore Cluster

2

Cluster Size

3

Review

Restore Mode

Restore From Region

Select the region of the backup. By default it is the same as the current cluster.

Oregon (us-west-2)

Restore Mode

Basic Snapshot Restore

Point-in-Time Restore

secondBackupSet (2023-11-01 06:21:14 UTC+00:00)

Restore to New Cluster

Name

Create a unique name for your cluster

downstream

Port

TiDB port for connection

4000

Region

Select the region for the new cluster. By default it is the same as the current cluster.

Oregon (us-west-2)

Billing Calculator

TiDB Cost

Compute Cost

\$ 1.5639

TiKV Cost

Compute Cost

\$ 2.6496

Storage Cost

\$ 0.0771

TiKV Cost

\$ 2.7267

TiFlash Cost

Compute Cost

\$ 1.3708

Storage Cost

\$ 0.0846

TiFlash Cost

\$ 1.4554

Plan Cost

Dedicated Tier

Your dedicated cluster.

\$ 5.746 /hour

See the detailed [Node Price](#).

Cancel

Next

- Choose the resource, it should be identical to the primary TiDB Cluster and no need to change.

Restore Mode & Restore Cluster

Cluster Size

Review

Cluster Size

TiDB

TiDB is the computing node that aggregates data from queries returned from transactional or analytical stores.

Node Size

8 vCPU, 16 GiB

Node Quantity

2

TiKV

TiKV is the storage node that stores the online transactional processing (OLTP) data.

Node Size

8 vCPU, 32 GiB

Node Quantity

3

Node Storage

The storage size of TiKV cannot be decreased after creation.

50

100

4096

GiB

TiFlash (Optional)

TiFlash is the analytical storage node that replicates data from TiKV in real-time and supports real-time analytical workloads.

Node Size

Billing Calculator

TiDB Cost

Compute Cost

\$ 1.5639

TiKV Cost

Compute Cost

\$ 2.6496

Storage Cost

\$ 0.0771

TiKV Cost

\$ 2.7267

TiFlash Cost

Compute Cost

\$ 1.3708

Storage Cost

\$ 0.0846

TiFlash Cost

\$ 1.4554

Plan Cost

Dedicated Tier

Your dedicated cluster.

\$ 5.746 /hour

See the detailed [Node Price](#).

Cancel

Previous

Next

- Start to restore, it should take about minutes for TiDB Clusters less than 500 GB.

✓ Restore Mode & Restore Cluster

✓ Cluster Size

3 Review

Restore Mode

SnapshotsecondBackupSet (2023-11-01 06:21:14 UTC±00:00)

Restore to New Cluster

Cluster Namedownstream

Port4000

Restore From Regionus-west-2

Restore To Regionus-west-2

Cluster Size

	Node Size (vCPU + RAM)	Node Storage	Node Quantity
TiDB info	8 vCPU, 16 GiB		2
TiKV info	8 vCPU, 32 GiB	100 GiB	3
TiFlash info	8 vCPU, 64 GiB	200 GiB	1

Billing Calculator

TiDB Cost

Compute Cost\$ 1.5639

TiKV Cost

Compute Cost\$ 2.6496

Storage Cost\$ 0.0771

TiKV Cost\$ 2.7267

TiFlash Cost

Compute Cost\$ 1.3708

Storage Cost\$ 0.0846

TiFlash Cost\$ 1.4554

Plan Cost

Dedicated Tier

Your dedicated cluster.

\$ 5.746 /hour

See the detailed [Node Price](#).

Cancel

Previous

Restore?

- Then, you will get a new TiDB Cluster.

Clusters

Create Cluster

Cluster Name	Tier Type	Status	Cloud	Region	Version	Creation Time	Action
downstream	Dedicated	Creating	AWS	Oregon (us-west-2)		2023-11-02 07:35:21	...
Cluster0	Dedicated	Available	AWS	Oregon (us-west-2)	v7.1.2	2023-11-02 05:14:36	...

- And it will be restored automatically.

Clusters

Create Cluster

Cluster Name	Tier Type	Status	Cloud	Region	Version	Creation Time	Action
downstream	Dedicated	Restoring	AWS	Oregon (us-west-2)	v7.1.2	2023-11-02 07:35:21	...
Cluster0	Dedicated	Available	AWS	Oregon (us-west-2)	v7.1.2	2023-11-02 05:14:36	...

- Security Setting and connect to the new TiDB Cluster.

Security Settings

To connect, you need to set the root password and allowed IP addresses first.

Root password to access the cluster

For your security, displaying and copying previous password is disabled. Click "Regenerate" or enter directly to reset the password.

Root Password

.....<>

Regenerate

Copy

Copy and save your password in a secure location.

Add entries to your IP Access List

Only IP address you add to your IP Access List will be able to connect to your cluster.

Add My Current IP Address

Allow Access from Anywhere

IP Address

Enter IP Address

Description

Enter a description

Add to IP List

IP Access List

0.0.0.0/0

Description

Allow Access from Anywhere.

Remove

Cancel

Apply

```
mysql> # in the 'downstream' tidb cluster.  
Query OK, 0 rows affected, 1 warning (0.31 sec)
```

```
mysql> select * from tbl;
```

id	name	value	description
1	first	0.1	first tuple in test table tbl
2	second	0.2	second tuple in test table tbl
3	third	0.3	third tuple in test table tbl

```
3 rows in set (0.48 sec)
```

```
mysql> # in the primary tidb cluster 'Cluster0';  
Query OK, 0 rows affected (0.26 sec)
```

```
mysql> select * from tbl;
```

id	name	value	description
1	first	0.1	first tuple in test table tbl
2	second	0.2	second tuple in test table tbl
3	third	0.3	third tuple in test table tbl

```
3 rows in set (0.27 sec)
```

- Data validations.
 - Check all the schemas you have, such as databases, tables, indexes and etc.
 - Check all the tuple count of the tables you have, especially the most important ones.
- **Do validation and some make-up of the data that is not restored, refer to <2.5 Verification and Make-up>**

2.4 Changefeed

Since we have set the GC variable `tidb_gc_life_time=168h` in the primary TiDB Cluster, it should not push the `tikv_gc_safe_point` later than the Backup Set you choose to restore. Then, we could create the `Changefeed` from the TSO of that Backup Set. Refer to <[GC Overview](#)>.

Refer to [Sink to MySQL](#) and [Changefeed](#) for more information.

2.4.1 Before Changefeed

Before we create Changefeed, the User should take care of the following issues:

- Make sure that all tables have the UNIQUE KEY or Primary Key, refer to [TiCDC FAQs](#).
- Take great care of the DB account of the destination TiDB Cluster.
 - Which should have enough privileges to write data, refer to [TiCDC FAQs](#).
 - DDLs that are replicated to the destination will be execute under that DB Account. As such, the newly created table will belong to that DB Account.
 - The restored TiDB Cluster will not have the same DB accounts and passwords as the primary TiDB Cluster.
- Currently, TiDB Cloud only allows up to 100 changefeeds per cluster.
- Currently, TiDB Cloud only allows up to 100 table filter rules per changefeed.
- RCU calculation, normally 16 RCU should be enough for most applications. Refer to [Changefeed Billing](#) for pricing and performance information.
- The `CREATE SEQUENCE` DDL operation and the `SEQUENCE` function in TiDB. When the upstream TiDB uses `SEQUENCE`, TiCDC ignores `SEQUENCE` DDL operations/functions performed upstream. However, DML operations using `SEQUENCE` functions can be correctly replicated.
- When you replicate a wide table with a large single row (greater than 1K), it is recommended to configure the `per-table-memory-quota` so that $\text{per-table-memory-quota} = \frac{\text{ticdcTotalMemory}}{(\text{tableCount} * 2)}$. `ticdcTotalMemory` is the memory of a TiCDC node, and `tableCount` is the number of target tables that a TiCDC node replicates.

2.4.2 Steps to Create Changefeed

The steps to create Changefeed are:

- Go back to the primary TiDB Cluster, and click the 'Create the Changefeed'.

Cluster0	Changefeed	Create Changefeed				
Overview		Name	Destination	Status	Checkpoint Time	Create Time
Chat2Query		changefeed-1	MySQL	Running	2023-11-02 18:45:38	2023-11-02 16:16:43
Data						...
Import						
Changefeed						
Backup						

- Configure the destination of Changefeed.

Cluster0
Overview
Chat2Query
Data
Import
Changefeed
Backup
Monitoring
Diagnosis
Metrics
Alerts
Events

Create Changefeed

1 Configure Destination
2 Configure Replication
3 Configure Specification
4 Review

Destination

☐ Kafka
☒ MySQL
☐ Amazon S3
☐ TIDB Serverless

MySQL Endpoints

MySQL Endpoints

tidb.xbtezky71i.clusters.staging.tidb-cloud.com:4000

User Name
Password

root

.....

Cancel
Next

- Configure Replication, start the replication position from the TSO of the Backup Set which has been restored above.

Cluster0
Overview
Chat2Query
Data
Import
Changefeed
Backup
Monitoring
Diagnosis
Metrics
Alerts
Events
Calibrate Resource
Security
SQL Users

Create Changefeed

1 Configure Destination
2 Configure Replication
3 Configure Specification
4 Review

Table Filter

☒ I understand the data inconsistency risk and want to replicate all tables without unique keys, including new tables created in the future.

Filter Rules
Tables with valid keys
Tables without valid keys

**

Tables without valid keys in the listed tables indicate that they **DONT** have any primary keys or unique indexes. Without a unique identifier for the downstream system to handle duplicate events, data consistency may be compromised during replication

Start Replication Position

☐ Start replication from now on
☒ Start replication from a specific TSO
☐ Start replication from a specific time

445358994620416000

Cancel
Previous
Next

- The RCU of Changefeed decides how much resource it is going to use for the replication, 16 RCUs should be enough. Refer to [Changefeed Billing](#) for pricing and performance information.

Cluster0

Overview

Chat2Query

Data

Import

Changefeed

Backup

Monitoring

Diagnosis

Metrics

Alerts

Events

Create Changefeed

Configure Destination ☒ Configure Replication ☒ **Configure Specification** ☒ Review ☐

Changefeed Specification

TiDB Cloud measures the capacity of TiCDC Changefeed in Replication Capacity Units (RCUs). When you create a Changefeed, you can select an appropriate specification. The higher the RCU, the better the replication performance. The performance values are for reference only and might vary in different scenarios. You will be charged for these TiCDC changefeed RCUs. Learn more in [Pricing Details](#).

Specify the number of RCU used by this changefeed

16 RCUs

Changefeed Name

Give this changefeed a name to make it easier to find later

changefeed-1

\$2.0912/hour

Cancel Previous **Next**

- Double check the information and click the **Submit**.

Cluster0

Overview

Chat2Query

Data

Import

Changefeed

Backup

Monitoring

Diagnosis

Metrics

Alerts

Events

Calibrate Resource

Security

Create Changefeed

Configure Destination ☒ Configure Replication ☒ **Configure Specification** ☒ **Review** ☒

Replication Destination

Destination MySQL

MySQL Endpoint tidb.xtbe2kyj7li.clusters.staging.tidb-cloud.com-4000

MySQL Username root

Replication Settings

Table Filter *

Start Position FROM TSO 445358994620416000

Specification and Name

Specification 16 RCUs

Name changefeed-1

\$2.0912/hour

Cancel Previous **Submit**

Cluster0

Overview

Chat2Query

Data

Import

Changefeed

Backup

Monitoring

Diagnosis

Metrics

Alerts

Events

Calibrate Resource

Security

Changefeed

Create Changefeed

Name	Destination	Status	Checkpoint Time	Create Time	Action
changefeed-1	MySQL	Creating	-	2023-11-02 16:16:43	...

- Check whether data updated after the Backup Set can be replicated to the 'downstream'.

```
mysql> insert into tbl values(4, "forth", 0.4, "forth tuple in test table tbl");
Query OK, 1 row affected (0.27 sec)

mysql>
mysql>
mysql> insert into tbl values(4, "forth", 0.4, "^C;
;
^C
mysql> # The changefeed is creating, but not finished. We add a tuple above
Query OK, 1 row affected (0.26 sec)

mysql> # After the changefeed is created, it will become running.
Query OK, 1 row affected (0.26 sec)

mysql> insert into tbl values(5, "fifth", 0.5, "fifth tuple in test table tbl");
Query OK, 1 row affected (0.26 sec)

mysql>
mysql>
mysql> # After that, let us check whether all data after the safe point has been replicated to the 'downstream';
Query OK, 1 row affected (0.31 sec)

mysql> select * from tbl;
+----+-----+-----+-----+
| id | name  | value | description |
+----+-----+-----+-----+
| 1  | first | 0.1   | first tuple in test table tbl |
| 2  | second | 0.2   | second tuple in test table tbl |
| 3  | third | 0.3   | third tuple in test table tbl |
| 4  | forth | 0.4   | forth tuple in test table tbl |
| 5  | fifth | 0.5   | fifth tuple in test table tbl |
+----+-----+-----+-----+
5 rows in set (0.27 sec)

mysql> # The result here should be identical with the 'downstream';
Query OK, 0 rows affected (0.26 sec)
```

```
mysql> # This is the 'downstream', and the tuples of `tbl` is:
Query OK, 0 rows affected (0.19 sec)

mysql> select * from tbl;
+-----+-----+-----+-----+
| id    | name   | value | description                                     |
+-----+-----+-----+-----+
| 1     | first  | 0.1   | first tuple in test table tbl                |
| 2     | second | 0.2   | second tuple in test table tbl               |
| 3     | third  | 0.3   | third tuple in test table tbl                |
| 4     | forth  | 0.4   | forth tuple in test table tbl                |
| 5     | fifth  | 0.5   | fifth tuple in test table tbl                |
+-----+-----+-----+-----+
5 rows in set (0.18 sec)
```

- You may update some more data and see whether the change can be replicated to the 'downstream'.

2.5 Verification and Make-up

Before we switch the Application to the new TiDB Cluster, we'd better have some verification and test.

The User should do the following works before the Application Switch:

- Verify and make-up the **DB accounts** in the restored TiDB Cluster, while the DB account from the primary TiDB Cluster will not be restored with *the TiDB version earlier than **6.2.x***.
- Verify and make-up the **SQL Binding** from the primary TiDB Cluster, which is not restored and replicated.
- Verify that **all the schemas** are identical between two TiDB Clusters if no DDL happens after the restore operation, such as databases, tables, indexes and etc.
- Verify that the **replication** is working in the restored TiDB Cluster, including both DMLs and DDLs. All data should be identical with the primary TiDB Cluster.
- Verify that all **global variables** are identical between the two TiDB Clusters. If not, fetch the value from the primary and set it to the restored TiDB Cluster.
- Verify and make-up the **SQL Binding** from the primary TiDB Cluster, which is not restored and replicated with *the TiDB version earlier than **6.2.x***.

- Verify and make-up the **statistical data for Optimizer** in TiDB, i.e., rebuild the system table `mysql.stat_*` for Optimizer to make good plan for queries.
- Verify and make-up the **auto-incremental cache** in the restored TiDB Cluster, which will be conflict with what is replicated from the primary TiDB Cluster.
- Verify and make-up the **placement rule** in the restored TiDB Cluster.

2.5.1 DB Accounts

With ***the TiDB version earlier than **6.2.x*****, all DB accounts in the primary TiDB Cluster will **NOT** be restored in the new TiDB Cluster. The User should have to restore each DB account by themselves.

And the `root` account in the new TiDB Cluster has a new password and different privileges from the `root` in primary TiDB Cluster.

The User should take good care of the DB accounts manually, if *the TiDB version is earlier than **6.2.x*.**

2.5.2 All Schemas

The User should fetch all the informations below from the primary TiDB Cluster, and check it in the restored one:

- Databases
- Tables
- Indexes
- Sequence
- Other objects in TiDB

2.5.3 Replication Validation

The validation should cover:

1. Do some data changes, as well as some DDL, to see whether such changes will be transferred to the downstream.
2. All schemas should be identical between the primary and downstream, as well as the incremental part.
3. All tuple counts of each table should be identical between the primary and downstream, as well as the incremental part. If there is any small difference, we should check the latency of the `Changefeed`.

Refer to :

- [TiCDC FAQs](#)

2.5.4 Global Variables

- Get all variables from the two TiDB Clusters, for example:

```
1 show variables;
```

- Compare all variables from the two TiDB Clusters.
- If there were differences, fetch the variables and set them into the restored TiDB Cluster. For example:

```
1 set global variables tidb_gc_life_time=10m;
```

Refer to:

- [<SHOW \[GLOBAL|SESSION\] VARIABLES | TiDB SQL Statement Reference>](#)
- [<SET \[GLOBAL|SESSION\] <variable> | TiDB SQL Statement Reference>](#)

2.5.5 SQL Binding

With ***the TiDB version earlier than **6.2.x*****, the SQL Bindings will NOT be restored automatically.

- Get all SQL Bindings from the two TiDB Clusters

```
1 show global bindings;
```

- Set in the downstream, refer to [SQL Plan Management \(SPM\)](#) .

```
1 CREATE [GLOBAL | SESSION] BINDING FOR BindableStmt USING BindableStmt
```

2.5.6 Analyze for `mysql.stat_*`

Since the `mysql.stat_*` will not be restored, we need to recover such data by running `analyze` command in the case of bad plan for queries, refer to [ANALYZE | TiDB SQL Statement Reference](#).

The process can be :

- After the downstream TiDB Cluster is restored, we'd better restore the `mysql.stat_*` by running `analyze $TableName` .
- Divide all tables into three different parts of importance:
 - Key tables
 - Major tables
 - Normal tables
- Fetch health status of all tables by `SHOW STATS_HEALTHY` , refer to [SHOW STATS_HEALTHY](#).
 - If there are `Key tables` that are not healthy, run the `analyze` immediately.
 - If there are `Major tables` that are not healthy, run the `analyze` once a day.
 - If there are `Normal tables` that are not healthy, ignore it or run the `analyze` once.

2.5.7 Auto-incremental ID

If the TiDB is older than 6.5.x, then you need to take care of the `cache` of `auto-incremental ID` downstream.

TiCDC will not change the `cache` of `auto-incremental ID` downstream. If there is `auto-incremental` column in a table as an ID, then there will be conflict after Application switched to the downstream TiDB Cluster.

The work-around is to reset the `cache` of `auto-incremental ID` downstream. Steps are:

- Fetch schemas from the primary TiDB Cluster to check whether there is `AUTO_INCREMENT` .

```
1 CREATE TABLE IF NOT EXISTS t1 (  
2   id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
3   b VARCHAR(200) NOT NULL  
4 );  
5  
6 show create table t1; // Then, grep AUTO_INCREMENT
```

- Then, **reset the `auto_increment` in the downstream before Application Switch:**

```
1 alter table xx auto_increment=0;
```

- Or, **just restart the TiDB Cluster before Application Switch.**

2.5.8 Placement Rule

The Placement Rules will not be restored, we have to restore it by rebuilding it, refer to [Placement Rules in SQL](#) .

2.6 Application Switch

1. Make sure that all steps above have been done, especially all the verifications and make-up.
2. Run 'sanity test' or 'integration test' of the Application in the new restored TiDB Cluster
3. If everything goes well, ***switch the database address of the online Application from the primary to the new one.***

2.7 Finish

Cleaning up operations:

- Make sure that the Application is working well in the newly restored TiDB Cluster.
- Backup the primary TiDB Cluster for emergencies.
- Delete the primary TiDB Cluster.
- If you don't want to delete the primary, remember to open the GC by :

```
1 set global tidb_gc_enable=on;
2
3 set global tidb_gc_life_time=10m;
```

3. Notes

After the backup is running and before the Application Switch:

- It would be better not to do DDL.
- It would be better not to run big transactions (long or a lot of changes).
- It would be better not to use the `auto-incremental` if TiDB is older than 6.5.0.
- It would be better not to use complex schemas.