Mizu VoIP Server –Database Failover From Backups

Contents

For VoIP service general availability, please refer to the VoIP High Availability guide.
For other database high availability methods, please refer to the HA Database guide.

This document discusses “failover from backups” method implemented by the Mizu VoIP Server.

Why and When?

The "**failover from backups**" is an easy to setup and low cost method to achieve high database availability for the Mizu VoIP server.

A few different method is available from the SQL engine to implement high availability as discussed here, such as Always-On, Clustering and Log Shipping. The problem with these methods is their high cost (additional hardware costs, extra CPU time), complicated setup and increased administrative costs due to non-trivial maintenance.
Hardware failures happens usually one time per 10 years in average and in case if your business can afford some data loss and service outage at such hardware failure events then we recommend the failover method described in this document.

The recommended target group for this model is medium sized VoIP service providers with up to 500 000 users and up to 20 000 simultaneous calls.
Using the failover model described in this document (backup based failover) results in some service outage and data loss on main database hardware failure, thus it is not recommended for critical infrastructure or critical applications such as transport and medical applications.
In case if your use-case requires better availability, we recommend to use Always-On Availability provided by the database engine.
In case if your use-case requires less availability or you are using only one server, then we recommend to just configure backups and manually restore in case of a failure.

Implementing this method will add fully automated restore capability for your VoIP server database, thus highly improving your service availability. While this method implies some service and data loss in case of failure, the low setup cost justifies its usage rather than relying only on database backups, which would require a manual restore and switchover in case of a failure.

The main characteristics of the backup based failover are the followings:

- Easy setup and low administration/maintenance needs
- Low hardware cost (you only need two servers and even these servers can be used also for other purposes if your traffic is low)
- A small amount of data can be lost in case of a hardware failure: usually a few minutes for important data (CDR's, billing, users) and a few hours for non-important data (settings, statistics).
- Some service outage in case of a hardware failure: usually around 20 minutes, depending on the configured failover wait time and the time needed to restore the backups to the secondary database.

Typically a database failover event should occur very rarely, thus the above described losses are affordable for most use cases

How it works?

Instead of costly clustering, log shipping or replication this method is based on simple foolproof backups thus are very easy to maintain and requires only very low extra hardware resources.
The automated backups have to be set from the MS SQL server and a few settings to be set in the Mizu VoIP server.
In this way you will have 3 types of backups:

- Full database backup created monthly by the database engine
- Differential backup created daily by the database engine
- Table level backups created continuously by the Mizu VoIP server (changes in tb_users, tb_cdr and other tables).

On database failure, the app service(s) will connect to the secondary server, will restore it from the last backups and continue the operation as normally.

By default the auto failover will happen only after around 10 minutes and multiple retries (configurable by the wait setting) in order to avoid unnecessary failovers such as outages triggered by an OS reboot.
The Mizu VoIP server will find all the backup files in the configured backup directory and automatically selects the latest one to be restored and then will apply the table level file backups from the date of the last full/diff backup.

Hardware requirements

For high availability, you will need at least two servers.
These can be configured as active-active or active-passive.
Regardless of the number of servers used, you can only have one single database server and multiple app servers. A stronger database server can serve tens of app servers.
You can also run an app service on the same server running the database service.
The following hardware configuration is for 200 000 users with 20 000 simultaneous calls.

- Database servers: 16 core CPU, 64 GB RAM, 500 GB fast (SSD) disk, 1 TB slower disk.
- App servers: 8 core CPU, 8 GB RAM, 60 GB slower disk

Configuration

In short:
All you have to do is to create periodic database backups (such as daily diff and monthly full), let the VoIP server to know about the backup location (by setting the externalbackupdir setting) and setup a secondary database server in the mserver.ini file under the [database2] section.

Details:
You will have to perform the configuration at one or more of the following places:

- MS SQL Management Studio (SSMS) Maintenance plans or using the BACKUP T-SQL (used to create scheduled periodic database backups)
- VoIP server config (.ini) file. This file can be found in the VoIP server app folder and its file name is usually mserver.ini. (used to define active and secondary servers and additional failover related settings)
- Configuration Wizard: from the MManage VoIP admin client -> Config menu -> Configuration wizard (used to set the most important backup related settings on the Maintenance page)
- VoIP server global config: the tb_settings table which can be managed from MManage -> Configurations form (used to define advanced backup and failover related settings)

We recommend the following configuration:

1. **Backup server**
   To achieve database high availability, you will need at least one backup server (often referred as “secondary” server) to host your backup database (passive backup).
   The database engine must be the same version (such as MS SQL 2019) and the database name must be the same as the primary database (default is “mserver”).
   You can also use the same server as an app server. In this case just setup a secondary service or clone the primary.
   The app servers can be configured as active-active or active-passive. You might add as many app servers as you wish (often referred as server instances or nodes).

2. **Backup location**
   It is important to store the backups on other server then your active database. Usually it should be stored on the secondary database server: Create a shared folder on your secondary/backup database and map as a network drive on your primary/active database server, then configure the backups to be stored in this folder.

   You need to also set the backup location for the VoIP server so it will know the path from where it has to restore the database in case of a failover. This can be done by from Configuration Wizard -> Maintenance page -> "Backup path" box or with the externalbackupdir global config.

3. **Configure monthly full backup and daily differential backup**
   You can easily configure backup from SMSS (SQL Management studio) by creating a new maintenance plan as described here and here.
   Alternatively the backups can be made also by the VoIP server. You can configure it from Configuration Wizard -> Maintenance page or with global settings.

4. **Configure db table file backup for the VoIP app server**
   File backups are used to recover more important data after the last database backup have been created to avoid data loss and enable rare database backups, thus saving precious hardware resource.
   The configuration can be done by from Configuration Wizard -> Maintenance page -> "Perform also file backups" checkbox or with the following global config options:
   - dbmaint_mssqlfilebackup: 1
   - dbmaint_filebackuplevel: 3
   - dbmaint_backupdbfiledir: set to your backup folder path (the mapped drive path mentioned above)

   More advanced backup related global config options can be found here.

5. **Configure failover in the ini file**
   All you need to do is to configure [databaseX] sections in the mserver.ini file for both the active and the secondary database server.
   Instead of using a single [database] section, you will need to create [database1] for the current active database and [database2] for the secondary database like this:

```
[database1]
location=192.168.1.11,2223
name=mserver
username=sa
password=xxx
node=X

[database2]
location=192.168.1.12,2223
name=mserver
username=sa
password=xxx
node=X
```

   More advanced ini file settings can be found here.

6. **Ready**

That's all. Backups are created now automatically and your server will failover to [database2] if [database1] will become unavailable as described here. See the below chapters for more advanced configuration and administration.

Scenarios

The failover procedure can be used with any number of servers, however in this section we discuss only failover possibilities with two servers for simplicity. Similar behavior is expected with more servers involved.
Terms:

- Server: individual whole server hardware or virtual machine with its own operating system
- Database: DB server engine running on a Server (MS SQL server service)
- App service: VoIP app server engine running on a Server (The Mizu VoIP server service)
- Backups: backup path to store database and other backups (usually a shared folder)

The failover method described in this document can be used with any setup, such as:

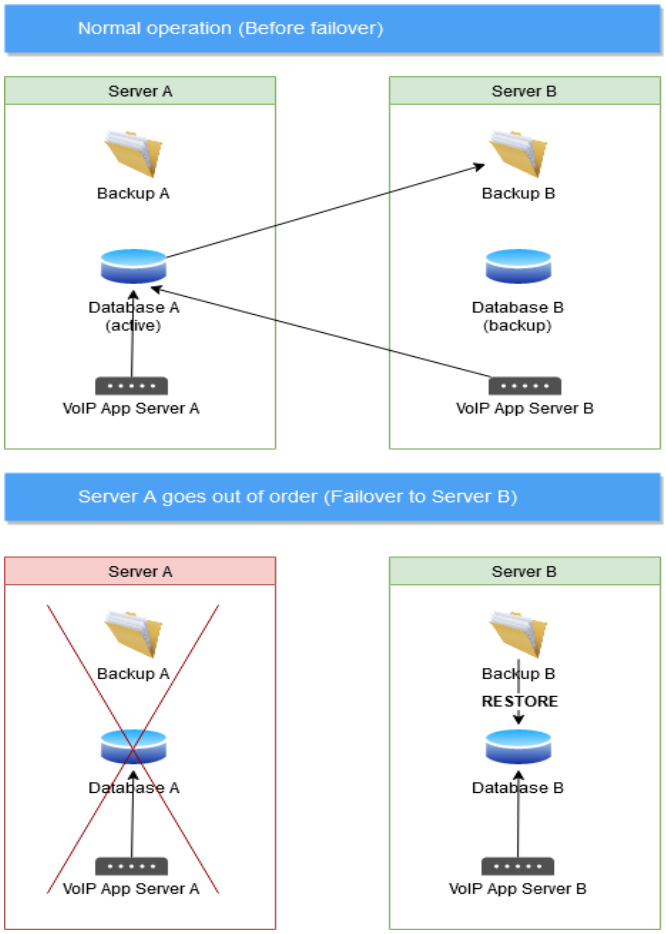
- 2 servers hosting both the database (active-backup) and the app service (active-active or active-passive)
- 3 servers: server A hosting the primary active database and app services A, server B hosting the backup database and app service B, server C hosting the app service C
- X servers: server A hosting the primary active database (and maybe an app service), server B hosting the backup database (and maybe an app service) and other servers hosting additional app services

Let's suppose that we have two servers:

- Server A hosting the active database A, app service A and backup location A.
- Server B hosting the backup database B, app service B and backup location B.

Server A fails

This means a hardware or OS level failure.
The app service B will to do the failover to database B.
To-do:
You might need to adjust your DNS SRV records or if you use a load balancer then it should automatically detect the problem and send all traffic only to App service B.
Restore server B and adjust the config as described at the maintenance section.



Server B fails

This means a hardware or OS level failure.

Nothing special will happen as server A will be able to continue to operation as-is.

To-do:

You might need to adjust your DNS SRV records or if you use a load balancer then it should automatically detect the problem and send all traffic only to App service A.

Restore server B so backups can resume and to have a working backup if server A fails.

Database A fails

(The hardware or OS is still OK, but the database engine fails or the database becomes corrupted)

The app service A or B will to do the failover to database B.

To-do:

Restore server B and adjust the config as described at the maintenance section.

Database B fails

(The hardware or OS is still OK, but the database engine fails or the database becomes corrupted)

Nothing special will happen as both app service A and B will be able to continue to operation as-is.

To-do:

Restore database B to have a working backup if server A fails.

App service A fails

(Application level error)
Nothing special happens as App service B will remain unaffected. You might need to adjust your DNS SRV records or if you use a load balancer then it should automatically detect the problem and send all traffic only to App service B.
To-do:
Fix the problem (fix the files / fix the configuration / restore old configuration / reinstall the service)

App service B fails

(Application level error)
Nothing special happens as App service A will remain unaffected. You might need to adjust your DNS SRV records or if you use a load balancer then it should automatically detect the problem and send all traffic only to App service A.
To-do:
Fix the problem (fix the files / fix the configuration / restore old configuration / reinstall the service)

Advanced configurations

Backup by the VoIP service

Instead of configuring the backup from MS SQL Management studio or with the BACKUP statement, you can schedule backups from the VoIP server. This can be configured from the Configuration Wizard Maintenance page by selecting the "Perform regular mssql backup" checkbox or with the following global config options:

- dbmaint_do: true (default is true, so you just have to make sure that it is not set to false)
- dbmaint_mssqldbbackup: 2 (-1: auto, 0=no,1=yes)
- dbmaint_backuplevel: 1 (0=no backups,1=daily,2=daily,monthly,weekly,3=hourly,daily,monthly,weekly,4=full,5=keep files)

Usually we recommend to set the dbmaint_mssqldbbackup parameter to 0 and schedule a backup from MS SQL Management Studio -> Maintenance Plans.

MS SQL Backup settings

Instead of monthly full and daily diff backup, you might use other scheduling, such as a full backup every week and a diff backup every hour if you wish to achieve lower data loss in the expense of some more hardware resources. The monthly full and daily diff is our well tested recommended scheduling.

The VoIP server should automatically find the backups, however you might use the following conventions to help this process:
We recommend to store the full and differential backups in different sub-folders using the following folder names:

- For full backup: full or yearly/monthly/weekly
- For differential backup: diff or hourly/daily

In case if you wish to store full backup in the daily folder, set the dailybackupisdiff global config to true for the VoIP server.
Instead of using the special keyword as the folder name, you might use the same strings as part of the backup file name.

During the failover database recovery, the VoIP server will automatically find the latest differential and full backup file to restore from.

Map network drive

To share a folder or map a network drive from your backup server on the active server, you need to have file sharing enabled in Windows.

You can simplify the administration by using the same account on both servers. For example the Administrator account with the same password if your server is not under active directory.
Make sure to use the same account for the service doing the backups (MS SQL Agent or the mserver service).
If fore some reason you can't use windows file sharing, then you can mount an FTP drive as described here and here.
You might also map the drive at runtime by using the run_before_backup global config to launch an external batch file which can contain something like this:
net use M: \\BACKUPSERVER\SHAREDFOLDER /user:USERNAME\PASSWORD
or
ftpuse M: 192.168.1.13/PATH PASSWORD /USER:USERNAME /PORT:21

Multiple app servers

In case if you have a failover database, most probably you are also using multiple app service (named server instances or nodes).
In this case make sure to adjust the ini file for each servers to contain the same values.
Also the backup paths have to be set differently on all servers (the externalbackupdir and/or the dbmaint_backupdbdir configs).

The failover will be handled by the "first" server which can't connect and the wait expires and for the recovery it will put a lock on the database, so there will be no conflicts between the app server trying to restore the same database.
To setup a failover for app servers (thus achieving high availability for your whole platform) please refer this guide.

Active – passive setup

There is no much difference between active – active and active – passive server configurations.
The VoIP service (node=2) must also be running on the passive server, the only difference is that in this case it will not handle any VoIP traffic by default, just acts as a failover agent: in case if the other server becomes unavailable, it will perform the failover and the database restore and it can begin handling the VoIP traffic.
The failover process is described here.

More database servers

Although two database (one active and one backup) usually covers all needs, you can use as many failover database servers as you wish.
Just create [databaseX] sections like [database1], [database2], [database3].
In this case the server will always try to failover to the "next" database if the current one is not available.
For example if the current active database is database1 and database2 is also unavailable, then the server will try to failover to database3.

Ini file configuration

The failover process can be fine tuned with ini file settings (mserver.ini file) in the [dbfailover] section.
All settings has meaningful/optimal defaults, however you are free to change these as you wish to better suit your requirements.

```
[dbfailover]
#set to 0 to disable failover or 1 to enable
enable=0
#specify wait time in seconds before to failover
wait=1080
#current active database (default is 1). You might need to explicitely set this if the active database is not at [database1]
activedb=1
#restore backup database before failover (0=no,1=yes,2=must)
restore=1
```

Details about the wait parameter:
When the app server can't connect to the database, it should not failover immediately but retry it multiple times until the wait period expires and during this time usually it also restart itself to filter out all app service issues.

- 1: immediately on any connection failure (special value to be used only for failover testing)
- below 60 (1 minute): very fast, without to try app server reset first (will result in unnecessary failovers. not recommended.)
- below 120 (2 minutes): very fast (might result in unnecessary failovers. not recommended.)
- below 360 (6 minutes): faster (might result in unnecessary failovers. recommended for aggressive failover.)
- below 1200 (20 minutes): normal (a good balance between downtime minimizations and avoiding unnecessary failbacks. recommended default.)
- above 1200 (20 minutes): slow (very conservative, with low chance for unnecessary failovers but it takes more time before to failover. recommended if you can afford a bit more downtime to make sure that you don't have to spend a lot of manual work restoring an unnecessary failover)

Default value is 1080 (18 minutes).

The following values might be created/set automatically by the VoIP server and should not be changed:

- restoreinprogress
- connectfail
- lastfailovertry
- failednewdb
- oldactivedb
- failoverresult
- dailybackupisdiff
- backupdbdir
- backupdbfiledir
- backupappfiledir
- externalbackupdir
- externalfullbackupdir
- externaldiffbackupdir

Global configurations

The most important backup related settings can be configured from the Config Wizard -> Maintenance page, however you might also change advanced configuration settings by the global config parameters from the Configurations form.
Search for these settings using the quick search box or list all backup related settings form MManage -> Config menu -> Configurations -> Security -> Backup.
The following backup related global config parameters are defined:

- dbmaint_do: set to true to enable database maintenance. default is true.
- dbmaint_backupdbdir: the backup folder to be used by the VoIP server for database backups
- dbmaint_backupdbfiledir: the backup folder for db table file backups (default is dbmaint_backupdbdir)
- externalbackupdir: external backup folder (the backup folder used by ms sql management studio)
- externalfullbackupdir: external backup folder for full backups (otherwise searched at externalbackupdir)
- externaldiffbackupdir: external backup folder for differential backups (otherwise searched at externalbackupdir)
- dbmaint_backuplevel: 0=no backups,1=daily,2=daily (default),monthly,3=hourly,daily,monthly,weekly,4=full,5=keep lots of files
- dbmaint_filebackuplevel: -1=auto (usually will default to 3), 0=no backups,1=daily,2=daily,monthly,3=hourly,daily,weekly,monthly,4=10 minute,5=extreme
- dbmaint_mssqldbbackup: specify if the VoIP server has to do database backups: -1: auto (default), 0=no,1=yes, 2=must
- dbmaint_mssqlexternaldbbackup: external db backup from SMSS if any: -1: auto (default), 0=no,1=yes, 2=must
- dbmaint_mssqlnodbbackup: set to 1 if there is no any backup configured
- dbmaint_mssqlfilebackup: table backups to file: -1: auto (1 if path is set to a different drive), 0=no,1=yes

- dbmaint_removedbfilebackup: remove database file backup after X days. default is 4
- dbmaint_dailybackupisdiff: set to true if backups in the "daily" folder are differential backups or to false if they are full backups
- backupappfiledir: app files backup folder path (default is dbmaint_backupdbfiledir)
- appfilebackuplevel: -1=auto, 0=nobackups,1=daily,2=daily,monthly,3=hourly,daily,weekly,monthly,4=10 minute
- appfilebackup: -1: auto (cert files no multiple nodes and has failover), 0: no, 1: cert files, 2: all files (2 is not recommended)
- removeappfilebackup: remove app file backup files after X days. -1: auto, 0: no, positive: days
- run_before_backup: external bat or exe to be launched once before backup
- run_before_failover: external bat or exe to be launched before failover
- run_after_failover: external bat or exe to be launched after failover

Limitations

- The failover process will take place only around 20 minute later after the database becomes unavailable (configurable by the wait parameter). During this time your VoIP service will be not available.
- Database restore on failover might take some time and during this your VoIP service is not available. Make sure that database restore can be performed within two hours, otherwise the failover process might become unreliable. This mostly depends on the database size, the disk speed of your backup database server and the read speed of the backup location.
- Some data loss might occur, especially non important data such as settings changes. This depends on the database backup interval and the time when the failure occurs. Make sure to recreate all such settings if necessary such as new routing entries added meantime.
- If failover fails, it will not try again for 4 hours.

Manual failover

Instead of automatic failover, you can also easily perform the failover manually.
If your primary server (or only the primary database) fails, restore the last backups to your secondary server and start the secondary app service.
You might have to rewrite the mserver.ini file for the app service(s) to point to the new database and you might have to reconfigure your domain name DNS records in case if you were using a single app server and the traffic needs to be redirected to the new one.
In case if you are using multiple app services then you might have to set the mainnode to a active service (if it was configured to the failed one; although this is also detected by the app services but only with a 3 days delay).

CLI

The following commands are defined and can be sent via the CLI, server console or API:

- dbbackup
  Will create a database backup.
  Syntax: dbbackup,reserved,path,ival,type
  Parameters:
    - reserved: not used
    - path: can be set to a path to backup to, otherwise it will backup to the default paths. If set to “def” then only default settings are used. Default is empty.
    - ival: how to take the backup: 1: hour, 2: day, 3: week, 4: month. Default is 2.
    - type: 0: normal, 1: must, 2: test
  Examples:
    - normal: backupdb
    - force: backupdb,,2,1
    - test: backupdb,,2,2

- dbrestore
  Will restore database from backup.
  Syntax: dbrestore,reserved,path
  Parameters:
    - reserved: not used
    - path: Specify the path to restore from or use empty if to restore from default (configured) location.

  Examples:
    - normal: dbrestore
    - from a specific path: dbrestore,,C:\mybackup
- dbfailover
  Will perform a database failover. (The same what the server would do if current/active database would become unreachable, so you can test db failover also by closing the current database service)
  Syntax: dbfailover,reserved,forcedbnum,checkonly,withrestore,needrestart
  Parameters:
    - reserved: not used
    - forcedbnum: Specify to which db to failover to (the X number in the [databaseX] ini file sections). Default is -1 which means auto detect next db to use.
    - checkonly: report only if failover is possible. Default is false.
    - withrestore: restore from last backup. Default is false.
    - needrestart: selfrestart after failover. Default is true

  Examples:
    - normal: dbfailover
    - with specific settings: dbfailover,,false,true,true

Maintenance

The most important thing: **make sure that you have backups!**
Once you configured the failover related settings described above, you should ensure that the backups are actually created and your server database is recoverable.
You might set automated email notifications to alert you if backups are missed. For example you might create a script to monitor your backup folder and alert if there is no any recently created file.

To **find out if your app service failovered**, just check the init file if the activedb value have been changed.
This should be also obvious from the server logs and the server will also try to notify you by email if email alerts are enabled.

In case if you need to **stop your main database intentionally**, you should stop the app service first or set the enable key to 0 in the ini file [dbfailover] section. However the default wait time is set to 1080 seconds (18 minutes) which should be enough even for a server restart.

In case **if a failover event occurs**, make sure to allow enough time for the database to recover before touching anything and you should also verify if your new database is in a good state once the recovery and failover is ready.
After a failover, you should restore or recreate your old database or setup a new database server and adjust the database backup paths to a folder on your new database server.
If you are using two servers then their roles will be inverted: the old active server will become the backup server and the old backup server will become the active server
Take the first full backup asap from your new active server to the new backup server.

You should also restore the history database to the new db service (this is the _backup database usually named mserver_backup).

In case if your server performs an **unnecessary failover**, do the followings:

1. Stop the supervisor service if any (MServiceHost)
2. Stop your app server(s) (from MManage -> Control or from Services -> “mserver” -> Stop)
3. Delete the [database] section in the ini file or adjust it to point to the desired database.
4. Adjust the [dbfailover] section in the ini file(s): set the activedb value back to the old db and delete the followings keys: restoreinprogress, connectfail, lastfailovertry, failednewdb, oldactivedb.
5. Start the app server(s)
6. Start the supervisor service if any
7. Restore any newly created items from the backup database (such as CDR’s created during the time the app server was failovered)

Related

Softswitch webpage
Large scale VoIP webpage
High availability VoIP
Database failover
Admin guide