

Back up and restore databases

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/backup-restore-databases/1-introduction>

Introduction

Completed

Planning and implementing a robust policy for data recovery is essential to mitigate the risks associated with user errors or technology failures. This involves exploring various options for backing up and restoring databases, including on-site and off-site solutions, cloud-based services, and automated backup systems. By understanding the different methods for data back up and recovery, organizations can safeguard their critical information and maintain operational continuity in the face of unforeseen challenges.

Even if you do nothing else to increase your availability with features like AGs or active geo-replication, you must have a solid backup and recovery strategy.

2. Back up and restore SQL Server running on Azure virtual machines

<https://learn.microsoft.com/en-us/training/modules/backup-restore-databases/2-backup-restore-sql-server-running-azure-virtual-machines>

Back up and restore SQL Server running on Azure virtual machines

Completed

SQL Server has two types of databases: system and user. System databases are the ones used by SQL Server such as master and msdb. User databases are the ones created by users that store the data for

applications. Both are important to account for when devising a backup and recovery plan. The nature of most system databases is that they're updated less frequently, although there are exceptions. As a general rule of thumb, system databases aren't restored from one SQL Server instance to another. Your main concern should be backing up the user databases.

The most common types of backups generated for SQL Server installations are full, differential, and transaction log. Depending on the deployment method, not all of these may be available as an option.

A full database backup is a backup of a single database. When the backup is made, all the pages from the database are copied to the backup device. The backup contains enough information so that you can restore the database to the point at which the backup was made. If you want to restore to a specific point-in-time to achieve your Recovery Point Objective (RPO), that can happen with the use of differential and/or transaction log backups. A full database backup backs up all the changes made to the database by the time the backup finishes.

A differential backup contains all the database pages that have changed since the last time a full backup was made.

A transaction log backup isn't only used to be able to achieve RPO and get to a more granular point in time but clears the transaction log and keeps its size manageable. Transaction log backups can be generated as frequently as every 30 seconds, although that is impractical.

Important

Understand how the transaction log works because it impacts not only how the transaction log is backed up, but also how you can do point-in-time recovery using transaction log files.

There are other backup options such as copy-only, file, filegroup, partial, and more.

A differential or a log backup can be restored after a full database is restored, as long as the database RESTORE command uses either the `WITH NORECOVERY` or the `WITH STANDBY` option. If neither option is used, the database RESTORE will do a recovery of the database, after which no extra backups can be applied.

Every SQL Server database uses one of three recovery models: FULL, BULK_LOGGED, or SIMPLE. The recovery model is set as a database option, and governs the type of backups and restores that could be used with the database. Most databases are set to FULL or SIMPLE. FULL allows all types of backups to be generated while SIMPLE doesn't allow transaction log backups. This means that if you have a smaller RPO, SIMPLE may not meet your needs as you can't restore to a granular point in time.

3. Back up a SQL Server virtual machine

Back up a SQL Server virtual machine

Completed

Azure Backup offers the capability to back up virtual machines (VMs) that host SQL Server. These backups contain not only the SQL Server databases but also all other contents within the VM, allowing for a complete restoration if needed. Although this approach may not be suitable for every scenario, it provides a robust safeguard against issues such as ransomware attacks.

VM-level backups are SQL Server-aware, also known as application-aware, ensuring the creation of application-consistent backups. This means that restoring a VM-level backup won't disrupt SQL Server functionality. When using this option, the SQL Server log indicates that the I/O has been momentarily frozen and then resumed upon completion.

Combining SQL Server backups with snapshots can potentially cause issues. If snapshot delays cause backup failures, set following value on the

`[HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\BCDRAGENT]` key:

`"USEVSSCOPYBACKUP"="TRUE"`

Use local disks or a network share for backup files

As with on premises SQL Server instances, databases can be backed up to disks attached to the VM or to network shares (including the file share in Azure called Azure Files) that SQL Server has access to. If you're backing up to disks local to the VM, ensure that they aren't written to the ephemeral storage that is erased upon shutdown or restart. You might also want to make sure that the backups are copied to a second location so as not to create a single point of failure.

Backup databases to and restore from URL

Another option is to configure backup to URL for the SQL Server instance installed in the VM. Unlike backups made on premises, backup and restore from URL for an IaaS VM is effectively a local option.

Back up to URL requires an Azure storage account and uses the Azure blob storage service. Inside the storage account, there are containers, and the blobs are stored there. Unlike a path on your local disk, the path to a backup file looks something like

`https://ACCOUNTNAME.blob.core.windows.net/ContainerName/MyDatabase.bak`. You can include more folder names under your container for easier identification of backups (for example, FULL, DIFF, LOG).

To back up to or restore from a URL, authentication must be established between the SQL Server instance and Azure. Remember that inside of an Azure VM, SQL Server doesn't know it's running on Azure. A SQL Server Credential can be composed of the Azure storage account name and access key authentication or a Shared Access Signature. If the former is used, the backup is stored as a page blob and if the latter, it's stored as a block blob. Starting with SQL Server 2016, only block blob is available so you should use a Shared Access Signature. From a cost perspective, block blobs are also cheaper, and Shared Access Signature tokens offer better security control.

Restoring from a URL is as simple as restoring from disk or a network share. In the SQL Server Management Studio UI, select URL from the backup media type in the Wizard. If using Transact-SQL, instead of using FROM DISK, you would use FROM URL with the appropriate location and backup file name. Here are some sample statements:

The following statement would back up a transaction log.

```
BACKUP LOG contoso  
TO URL = 'https://myacc.blob.core.windows.net/mycontainer/contoso202003271200.trn'
```

The following statement would restore a full database backup without recovering it, so that a differential or transaction log backups could be applied.

```
RESTORE DATABASE contoso  
FROM URL = 'https://myacc.blob.core.windows.net/mycontainer/contoso20200327.bak'  
WITH NORECOVERY
```

Automated backups using the SQL Server resource provider

Any IaaS VM that has SQL Server installed can use the SQL Server resource provider. One of its options is the ability to configure automated backups so Azure takes care of backing up SQL Server databases. It requires the use of a storage account.

One benefit of implementing backups this way is that you can manage retention times for the backups. Another benefit is that you can ensure RPO due to the ability to take database and transaction log backups all in one easy-to-configure place. The following image shows an example of what configuring an automated backup looks like in the Azure portal.

Automated patching

Set a patching window during which all Windows and SQL patches will be applied.

Automated patching ⓘ **Enabled**
Sunday at 2:00
[Change configuration](#)

Automated backup

Automated backup ⓘ

Retention period (days) 30

Storage account [Select Storage Container](#)

Encryption

Backup system databases ⓘ

Configure backup schedule ⓘ

Backup frequency ⓘ

Backup start time (local VM time) 02:00 ▼

Backup time window (hours) 2

Log backup frequency (minutes) 60

The automated backup option is currently only available for Windows Server-based SQL Server installations.

Important

You choose one method of backing up databases with IaaS-based SQL Server deployments. For example, if you use automated backups, especially with transaction log backups, don't also configure those at the instance level inside the VM. You could cause problems with the log chain with restoring a database if things are uncoordinated, because each log backup clears the log and you must have an entire unbroken chain of log backups in order to do a log restore. For example, if transaction log backups happen inside the guest and at the Azure level, you may have to piece together the backups to do a restore.

While the backups can be automated, restores can't be. You would need to configure and use the restore from URL functionality within SQL Server.

4. Back up and restore a database for SQL Database and SQL Managed Instance

<https://learn.microsoft.com/en-us/training/modules/backup-restore-databases/4-use-azure-sql-database>

Back up and restore a database for SQL Database and SQL Managed Instance

Completed

Back up and restore on SQL Server PaaS offering work differently than on IaaS. Backups are generated automatically for Azure SQL Database, and Azure SQL Managed Instance. A full backup is created once a week, a differential every 12 hours, and transaction log backups every 5 to 10 minutes. All backups are located in read-access, geo-redundant (RA-GRS) blobs replicated to a datacenter that is paired based on Azure rules. That means backups are safe from an outage in a single data center.

Database backup and restore for SQL Database

Database backups are an essential part of any business continuity and disaster recovery strategy because they help protect your data from corruption or deletion.

SQL Database can assist you to be compliant with mandatory backups for regulatory purposes with retention policies. Backup policies can be configured per database as shown in the following image:

Configure policies



SQL server

Point-in-time-restore

Specify how long you want to keep your point-in-time backups. [Learn more](#)

How many days would you like PITR backups to be kept? ⓘ

Differential backup frequency

Specify how often you want differential backups to be taken. [Learn more](#)

Take a differential backup every:

Long-term retention

Specify how long you want to keep your long-term retention backups. You may choose to keep yearly backups for up to 10 years. [Learn more](#)

Weekly LTR Backups

Keep weekly backups for:

Monthly LTR Backups

Keep the first backup of each month for:

Yearly LTR Backups

Keep an annual backup for:

Which weekly backup of the year would you like to keep?

Apply

Cancel

If the server containing the database is deleted, all backups are deleted at the same time, and there's no way to recover them. If the server isn't deleted but the database is, you can restore the database

normally.

Both SQL Database, and SQL Managed Instance have a feature called Accelerated Database Recovery (ADR). This feature is enabled by default, and its purpose is to decrease the time it takes to deal with long running transactions so they don't affect the recovery time. Although Accelerated Database Recovery was developed for Azure and was originally an Azure-based feature, ADR was implemented in SQL Server 2019 as well.

Note

You can't restore SQL Database Azure SQL Managed Instance backups on SQL Database.

Automated database backups are also available on [Azure SQL Managed Instance](#). SQL Server database engine backups are automatically managed by Microsoft, and stored on Microsoft-managed Azure storage accounts.

Point in time restore

To restore a database to a specific point in time on SQL Database, you can use either Azure portal, Azure PowerShell, Azure CLI, or REST API.

Home > SQL databases > AdventureWorks (contoso/AdventureWorks) > contoso > AdventureWorks (contoso/AdventureWorks)

Create SQL Database - Restore database

Microsoft

Basics Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription ⓘ My Subscription ▼

Resource group ⓘ contoso_rg ▼

Source Details

Select a backup source and details. Additional settings will be defaulted where possible based on the backup selected.

Source Database AdventureWorks

Select source Point-in-time ▼

Earliest restore point 2021-11-23 00:00 UTC

Restore point (UTC) * 11/30/2021 1:51:00 PM

i Choose a restore point between the earliest restore point and the current time in UTC.

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * AdventureWorks_2021-11-30T13-51Z

Server ⓘ contoso (West US) ▼

Want to use SQL elastic pool? * ⓘ ☐ Yes ☒ No

Compute + storage * ⓘ

General Purpose
Serverless, Gen5, 2 vCores, 32 GB storage
[Configure database](#)

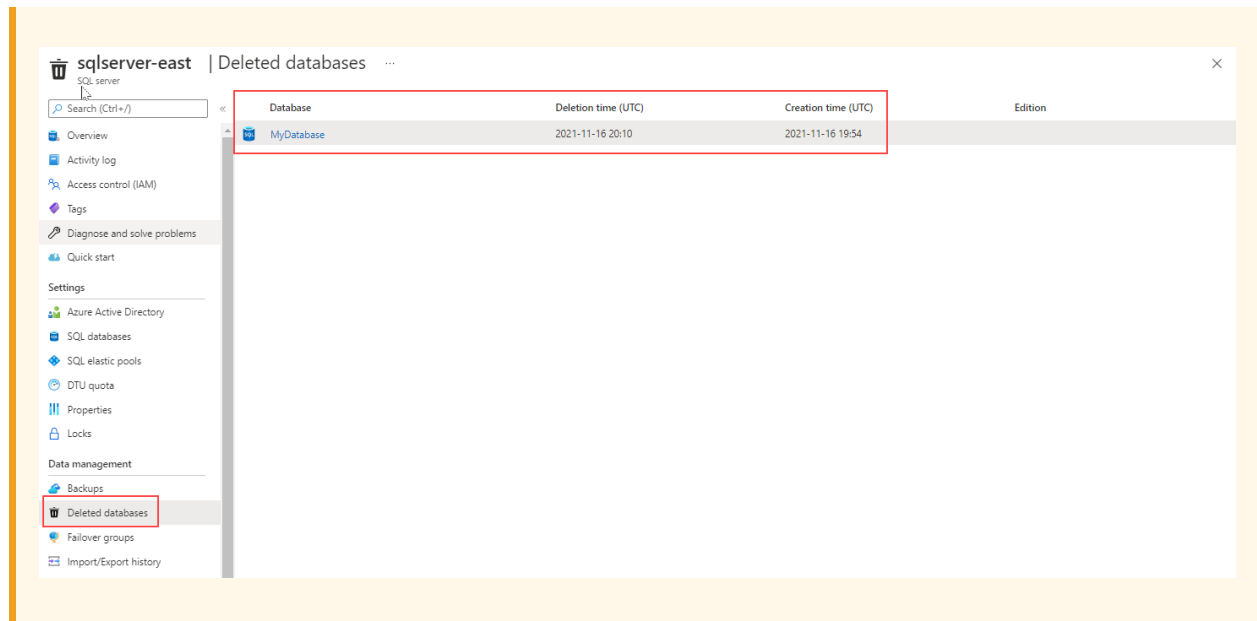
The following image shows the SQL Database restore page on Azure portal, where you can restore a database to a specific point in time.

Restore in place isn't supported on SQL Database, and SQL Managed Instance. You need to make sure the database doesn't exist before attempting the restore operation. By default, point in time retention policy is set to seven days, and you can change it to up to 35-days.

Restore a deleted database

Both SQL Database, and SQL Managed Instance have a feature to restore a deleted database to the last point in time available before the `DROP DATABASE` took place. To recover a deleted database to

the deletion time by using the Azure portal, open the server's overview page and select **Deleted databases**. Select a deleted database that you want to restore, and then enter the name for the new database that will be created with data restored from the backup.



The image shows how to restore a deleted database on SQL Database. The *deleted databases* page shows a list of deleted databases available to restore, the database deletion time in UTC, and the database creation time in UTC. Once you select the database, the *Create SQL Database - Restore database* page opens. On that page you find the earliest restore point in time available for the selected database.

Database backup and restore for SQL Managed Instance

Azure manages backups for databases in SQL Managed Instance automatically, and they operate similar to SQL Database.

You can also manually back up, and restore databases with SQL Managed Instance using the same backup to URL/restore from URL functionality found in SQL Server covered earlier. That requires the use of credentials to access the Azure Blob Storage container. SQL Database doesn't support this feature.

You can only generate a `COPY_ONLY` backup since SQL Managed Instance is maintaining the log chain. A sample backup statement would look like:

```
BACKUP DATABASE contoso
TO URL = 'https://myacc.blob.core.windows.net/mycontainer/contoso.bak'
WITH COPY_ONLY
```

Note

You can't restore SQL Database SQL Managed Instance backups on SQL Database.

5. Exercise: Backup to URL

<https://learn.microsoft.com/en-us/training/modules/backup-restore-databases/5-exercise-backup-url>

Exercise: Backup to URL

Completed

In this exercise, you'll learn how to perform a database backup to an URL.

Note

To complete this exercise, you'll need an [Azure subscription](#).

Launch the exercise and follow the instructions.

[Launch Exercise](#)

6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/backup-restore-databases/6-knowledge-check>

Module assessment

Completed

7. Summary

Summary

Completed

Without backups, you don't have high availability or disaster recovery. Features like AGs or active-geo replication don't replace a proper backup and recovery strategy. Backups must be tested via a restore otherwise your backups are just files that sit on a file system that checks a box somewhere that backups were generated.

Now that you've reviewed the module, you should be able to describe:

- Database backup and restore options for IaaS
- Virtual machine backup and restore options for IaaS
- Backup and restore options for PaaS