### *Database Backup/Restore Use Cases:*

* *Backups should not be on the same physical storage as your database files*
* *Manage disk space and archive previous backup files.*
* *Perform trial restorations.*
* *Secure devices and media.*
* *Set up schedules to automate your backups*
* *Use test server to verify the restore procedure of your backups*
* *Test your recovery strategies*
* *Use all available verification options during backup process*
* *Doing a FULL backup daily*
* *Doing differential backups more frequently*
* *Back up system databases as well*
* *Backup your most sensitive data to tape*

### *S3:*

*Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.*

### *EC2:*

*Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment.*

### *EBS:*

*Amazon Elastic Block Store (EBS) is an easy to use, high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS.*

### *VPC:*

*Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.*

### *AWS Direct Connect*

*is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.*

### *Amazon Relational Database Service (Amazon RDS)*

*It makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.*

*Amazon RDS is available on several*[*database instance types*](https://aws.amazon.com/rds/instance-types/)*- optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including*[*Amazon Aurora*](https://aws.amazon.com/rds/aurora/)*,*[*PostgreSQL*](https://aws.amazon.com/rds/postgresql/)*,*[*MySQL*](https://aws.amazon.com/rds/mysql/)*,*[*MariaDB*](https://aws.amazon.com/rds/mariadb/)*,*[*Oracle Database*](https://aws.amazon.com/rds/oracle/)*, and*[*SQL Server*](https://aws.amazon.com/rds/sqlserver/)*. You can use the*[*AWS Database Migration Service*](https://aws.amazon.com/dms/)*to easily migrate or replicate your existing databases to Amazon RDS.*

### *Amazon Redshift*

*is a fully managed, petabyte-scale data warehouse service in the cloud. You can start with just a few hundred gigabytes of data and scale to a petabyte or more. This enables you to use your data to acquire new insights for your business and customers.*

*The first step to create a data warehouse is to launch a set of nodes, called an Amazon Redshift cluster. After you provision your cluster, you can upload your data set and then perform data analysis queries. Regardless of the size of the data set, Amazon Redshift offers fast query performance using the same SQL-based tools and business intelligence applications that you use today.*

### *Amazon DynamoDB*

*is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.*

*Many of the world's fastest growing businesses such as Lyft, Airbnb, and Redfin as well as enterprises such as Samsung, Toyota, and Capital One depend on the scale and performance of DynamoDB to support their mission-critical workloads.*

### *Amazon Route 53*

*is a highly available and scalable cloud*[*Domain Name System (DNS)*](https://aws.amazon.com/route53/what-is-dns/)*web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.*

*Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS*

### *VPC peering*

[*Amazon Virtual Private Cloud*](https://docs.aws.amazon.com/vpc/latest/userguide/)*(Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined.*

*A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).*

*
            A VPC peering connection
        *

### *Amazon CloudFront*

*is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers’ users and to customize the user experience. Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don’t pay for any data transferred between these services and CloudFront.*

* *Backups should not be on the same physical storage as your database files*

*So, as you might suspect, having the database backup on the same physical drive as the database itself is not a particular good idea, because in case something bad happens to the drive you lose both your database and your backup.*

*This is why one of the first tips is to put your backups on a different storage, as far as possible, in terms of risk, from the same place where the database it backs up.*

*It might look like a simple and obvious thing to do, but in case you’re not doing it, you should, because it will save you a lot of work*

* + *Set up schedules to automate your backups*

*The next step in prevention and backup safety and reliability is to set up (automate) the backup process through the means of schedules.*

*Creating such schedules is important because, with the passing of time, your backups get older and out of date. Also, with time, there is a higher risk of data loss.*

*Keep yourself protected and be sure that you always have at hand a way to reestablish your data up until the point where the database failed.*

*These schedules will provide you an exact history of your data with the frequency you specify, depending on the business needs of your application or company, the importance of your data or other particular factors.*

*Use test server to verify the restore procedure of your backups*

*You might think your data is safe if the backup process has finished succesfully, but there is still another step that has to be done before you can safely say you have backed up your data, the restore process.*

*Without going through the entire process of recovering your data, through the restoring process, you cannot be sure that you can still access it. Other factors might influence this, like media corruption, and you cannot get hold of your data after backup.*

*Anoter item in the list of backup best practices is to frequently do a restoring test with all of the options you are going to use in a real-life scenario, on a test server. This will minimize the chances that you are caught off guard in the event of an unfortunate situation.*

* *Test your recovery strategies*

*The next bulletpoint in the backup best practices list is in regard to recovery strategies.*

*The success of a company is determined by its ability to quickly respond to changes. Although backup strategies are created to cover a full array of events, with time, they might become out of date.*

*This is why, periodically, you should take the time and go over all of the scenarios and see if there are any new elements that should be introduced in your specific strategy.*

*These scenarios should take into consideration both the impact on your individual database restores and also on the backups of your system databases.*

* *Use all available verification options during backup process*

*One more tip we can give about backup best practices is that you should be using all of the available verification options when you are doing your backups. This will help you be sure that all of the backups that you create will be created correctly and will be consistent from a transactional point of view.*

*Now, depending on how you prefer to do your backups, with T-SQL, through a wizard or other applications, there are multiple options.*

*For example, if you are using T-SQL:*

*Make sure that when you write your BACKUP command you also include the CHECKSUM condition in the WITH clause. This will force the operation to verify that each page of data written to the backup is passed through a checksum operation to ensure consistency on the backup media. A sample query would be like this:*

* + - *BACKUP DATABASE MainDB TO DISK = 'X:\Backups\MainDB.bak'*
    - *WITH CHECKSUM;*

*In the case you’re using a visual interface, for example SSMS (SQL Server Management Studio), make sure to have ticked the first two options in the Reliability section:*

* + - *The options, Verify backup when finished and Perform checksum before writing to media are going to act as an additional step of verification in the backup workflow and will guarantee that your backup is consistent, along with the data in it.*

*A different approach of making backups would be through SQL Server Maintenance Plans:*

* + - *Here you will have to tick a single checkbox, the option for Database Integrity.*

*If all of the above options seem to be for any reason out of hand for you, then we recommend using another piece of software, [SQLBackupAndFTP](http://sqlbackupandftp.com/). It is a very very simple tool to use and it has all of the most important options.*

*Now, we’re just over half-way through our list, and if you have started to incorporate any of these practices in your backup process, then you will surely have skipped yourself a few headaches in the future.*

*Now, continuing with our list, the next items in our backup best practices list are:*

* *Doing a FULL backup daily*

*This might sound as a bit too resource intensive operation, but it is the best way to prevent any data loss.*

*Now, depending on the business needs of your organization you should pick a recovery model (*[*more details here*](http://sqlbak.com/blog/sql-server-backup-and-restore/)*) that will be your organization’s first line of protection against data loss. If your organization can accept to lose 15, 30 or 60 minutes worth of data, then you should go for a simple recovery model.*

*Most cases would require you to do this FULL backup on a daily basis, but despite the overhead added by the time and occupied resources for such a process, the guarantee that your data is safe is, in our opinion, worth it.*

*If the organization you’re doing backups for has small databases with data that is easily recoverable, then a FULL backup should definetly be the correct way to go.*

*As we have already mentioned [SQLBackupAndFTP](http://sqlbackupandftp.com/) earlier, if you have been curious and downloaded it, you will be surprised to see that it has, by default, set its backup schedule to do a FULL backup every 24 hours.*

*This is very helpful as this way there is no more risk that a detail that is most important will be overlooked.*

*If you are to incorporate just this item from our list of backup best practices, then you have a great advantage as the recovery is easier if the backup is just one file and no database logs are needed.*

*Although, there is a downside to this as data can only be recovered up until the date when the backup was created and, depending on your database size, the backup can occupy a lot of disk space.*

* + *Doing differential backups more frequently*

*We just said that you should consider doing FULL backups on a daily basis, but even if you do not do it everyday, differential backups should be done even more frequently than FULL backups.*

*Differential backups are not so resource hungry and data loss is reduced even more if they are done on a frequent basis, after having done a FULL backup.*

*In terms of advantages, differential backups will take less time to complete and also less disk space as they will contain only the data that has been created since the last FULL backup. No database logs are needed in this case either and like we’ve said before, data will be more up to date.*

*The disadvantage of this backup type is also the fact that data can be recovered only up until the date when the last differential backup was created and the restore process of the data will be slightly more complicated as now there are two files that have to be incorporated into the restore process.*

* + *Do transaction log backups even more frequently*

*The following item in our list of backup best practices is one that leads the FULL and differential backups in terms of the frequency it should be made.*

*Because transaction logs actually contain all the recent activity that has occurred in the database, it can be used to make a restore of the database to a specific point in time, and this is its biggest advantage!*

*By doing a transaction log backup you also truncate the log and this way you keep it from becoming full. Also, transaction log backups can be performed while the system is working.*

*If the frequency of new data that is created in your database is very high, then we recommend doing a transaction log backup every 10 minutes, while for other databases which are less active doing such a backup every 30 or 60 minutes should be sufficient.*

*The downside of this backup type, though, is that the database has to have transaction logging activated which will increase the size of the database and will also increase the effort when doing a restore of a transaction log.*

* + *Back up system databases as well*

*You might think that having a backup strategy with all of the points mentioned above is more than enough to protect your organization from data loss, but it is not.*

*A backup strategy is incomplete if you do not have a backup plan for your SQL Server system databases, master, model and msdb.*

* + *Backup your most sensitive data to tape*

*This might not be a very common situation, but if your organization wants to be fully protected against data loss, then be sure to back up the data that it cannot afford to lose to tape.*

*Tape is much more resistant to physical shock than disk drives and due to this it has a major advantage. Just imagine a situation where you drop a box full of disk drives containing the last 3 months worth of data down the stairs. At that very second, you lose 3 months worth of data, but with tape drives you can pick the tapes up, and continue on.*

*Because there is no magnetic and nearly no electrical components in the drive, it uses less power than a conventional disk drive.*

*They are less sensitive to wear, as general media life for tapes are around 20 to 30 years and the lifespan of the drive is around 10 years.*

*So in conclusion, with tape you get:*

* + - *Very long media lifetime (20 – 30 years)*
    - *Very easy to transport*
    - *Lower power consumption*
    - *Higher reliability*
    - *A long lifetime for the drive (around 10 years)*