**RDS**

RDS is an Online Transaction Processing (OLTP) type of database.

The primary use case is a transactional database (rather than analytical).

Best for structured, relational data store requirements

Database instances are accessed via endpoints.

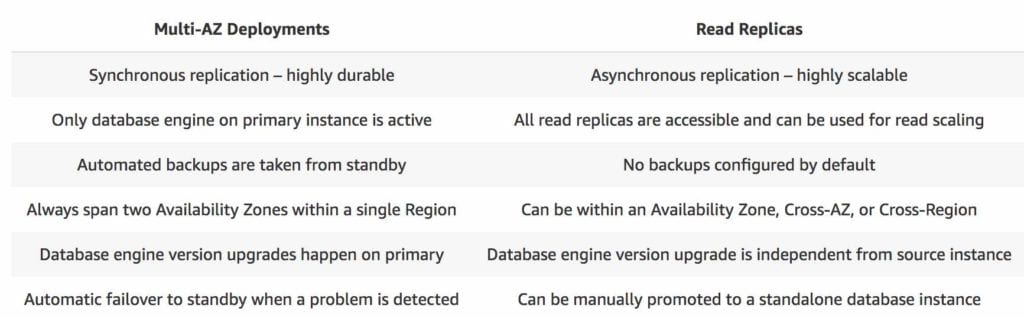
Endpoints can be retrieved via the DB instance description in the AWS Management Console, **DescribeDBInstances API** or describe–db–instances command.

You cannot encrypt an existing DB, you need to create a snapshot, copy it, encrypt the copy, then build an encrypted DB from the snapshot.

* A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same region.

If the master and Read Replica are in different regions, you encrypt using the encryption key for that region

* You can’t have an encrypted Read Replica of an unencrypted DB instance or an unencrypted Read Replica of an encrypted DB instance.



AWS recommends the use of provisioned IOPS storage for multi-AZ RDS DB instances.

It is recommended to implement DB connection retries in your application.

Recommended to use the endpoint rather than the IP address to point applications to the RDS DB.

The method to initiate a manual RDS DB instance failover is to reboot selecting the option to failover.

A DB instance reboot is required for changes to take effect when you change the DB parameter group or when you change a static DB parameter.

The secondary DB in a multi-AZ configuration cannot be used as an independent read node (read or write).

System upgrades like OS patching, DB Instance scaling and system upgrades, are applied first on the standby, before failing over and modifying the other DB Instance.

In multi-AZ configurations snapshots and automated backups are performed on the standby to avoid I/O suspension on the primary instance.

Amazon RDS Read Replicas for MySQL and MariaDB support Multi-AZ deployments

Q: Are there hybrid or on-premises deployment options for Amazon RDS?

Yes, you can run RDS on premises using Amazon RDS on Outposts and Amazon RDS on VMware. Please see the [Amazon RDS on Outposts](https://aws.amazon.com/rds/outposts/faqs/) and [Amazon RDS on VMware](https://aws.amazon.com/rds/vmware/faqs/) FAQs for additional information.

Q: How do I access my running DB instance?

Once your DB instance is available, you can retrieve its endpoint via the DB instance description in the [AWS Management Console](https://console.aws.amazon.com/), [DescribeDBInstances API](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_DescribeDBInstances.html) or [describe-db-instances command](http://docs.aws.amazon.com/cli/latest/reference/rds/describe-db-instances.html). Using this endpoint you can construct the connection string required to connect directly with your DB instance using your favorite database tool or programming language. In order to allow network requests to your running DB instance, you will need to authorize access.

Q: How many DB instances can I run with Amazon RDS?

By default, customers are allowed to have up to a total of 40 Amazon RDS DB instances. Of those 40, up to 10 can be Oracle or SQL Server DB instances under the "License Included" model. All 40 can be used for Amazon Aurora, MySQL, MariaDB, PostgreSQL and Oracle under the "BYOL" model. Note that RDS for SQL Server has a limit of up to 100 databases on a single DB instance to learn more see the [Amazon RDS SQL Server User Guide](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQLServer.html#SQLServer.Concepts.General.FeatureSupport.Limits).

Q: What should I do if my queries seem to be running slowly?

For production databases we encourage you to enable [Enhanced Monitoring](https://aws.amazon.com/rds/faqs/#Enhanced_Monitoring), which provides access to over 50 CPU, memory, file system, and disk I/O metrics. You can enable these features on a per-instance basis and you can choose the granularity (all the way down to 1 second). High levels of CPU utilization can reduce query performance and in this case you may want to consider scaling your DB instance class. For more information on monitoring your DB instance, refer to the [Amazon RDS User Guide](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Monitoring.html).

If you are using RDS for MySQL or MariaDB, you can access the slow query logs for your database to determine if there are slow-running SQL queries and, if so, the performance characteristics of each. You could set the "slow\_query\_log" DB Parameter and query the mysql.slow\_log table to review the slow-running SQL queries. Please refer to the [Amazon RDS User Guide](http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/Appendix.MySQL.CommonDBATasks.html) to learn more.

Q: Do I need to enable backups for my DB Instance or is it done automatically?

By default, Amazon RDS enables automated backups of your DB Instance with a 7 day retention period. Free backup storage is limited to the size of your provisioned database and only applies to active DB Instances. For example, if you have 100 GB of provisioned database storage over the month, we will provide 100 GB-months of backup storage at no additional charge.

If you would like to modify your backup retention period, you can do so using the console or the CreateDBInstance API (when creating a new DB Instance) or the the ModifyDBInstance API (for existing instances). You can use these APIs to change the RetentionPeriod parameter to any number from 0 (which will disable automated backups) to the desired number of days. The value can't be set to 0 if the DB instance is a source to Read Replicas

Q: Where are my automated backups and DB snapshots stored and how do I manage their retention?

Amazon RDS DB snapshots and automated backups are stored in S3.

Q: What happens to my backups and DB snapshots if I delete my DB instance?

When you delete a DB instance, you can create a final DB snapshot upon deletion; if you do, you can use this DB snapshot to restore the deleted DB instance at a later date. Amazon RDS retains this final user-created DB snapshot along with all other manually created DB snapshots after the DB instance is deleted. Refer to the [pricing page](https://aws.amazon.com/rds/pricing/) for details of backup storage costs.

Automated backups are deleted when the DB instance is deleted. Only manually created DB Snapshots are retained after the DB Instance is deleted.

Q: What is Amazon Virtual Private Cloud (VPC) and how does it work with Amazon RDS?

Amazon VPC lets you create a virtual networking environment in a private, isolated section of the AWS cloud, where you can exercise complete control over aspects such as private IP address ranges, subnets, routing tables and network gateways. With Amazon VPC, you can define a virtual network topology and customize the network configuration to closely resemble a traditional IP network that you might operate in your own datacenter.

One way that you can take advantage of VPC is when you want to run a public-facing web application while still maintaining non-publicly accessible backend servers in a private subnet. You can create a public-facing subnet for your webservers that has access to the Internet, and place your backend RDS DB Instances in a private-facing subnet with no Internet access.

Q: What is a DB Subnet Group and why do I need one?

A DB Subnet Group is a collection of subnets that you may want to designate for your RDS DB Instances in a VPC. Each DB Subnet Group should have at least one subnet for every Availability Zone in a given Region. When creating a DB Instance in VPC, you will need to select a DB Subnet Group. Amazon RDS then uses that DB Subnet Group and your preferred Availability Zone to select a subnet and an IP address within that subnet. Amazon RDS creates and associates an Elastic Network Interface to your DB Instance with that IP address.

Please note that, we strongly recommend you use the DNS Name to connect to your DB Instance as the underlying IP address can change (e.g., during failover).

For Multi-AZ deployments, defining a subnet for all Availability Zones in a Region will allow Amazon RDS to create a new standby in another Availability Zone should the need arise. You need to do this even for Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

Q: Can I move my existing DB instances from inside VPC to outside VPC?

Migration of DB Instances from inside to outside VPC is not supported. For security reasons, a DB Snapshot of a DB Instance inside VPC cannot be restored to outside VPC. The same is true with “Restore to Point in Time” functionality.

Q: What precautions should I take to ensure that my DB Instances in VPC are accessible by my application?

You are responsible for modifying routing tables and networking ACLs in your VPC to ensure that your DB instance is reachable from your client instances in the VPC.

For Multi-AZ deployments, after failover, your client EC2 instance and RDS DB Instance may be in different Availability Zones. You should configure your networking ACLs to ensure that cross-AZ communication is possible.

Q: Can I encrypt connections between my application and my DB Instance using SSL/TLS?

Yes, this option is supported for all Amazon RDS engines.

Amazon RDS generates an SSL/TLS certificate for each DB Instance. Once an encrypted connection is established, data transferred between the DB Instance and your application will be encrypted during transfer.

While SSL offers security benefits, be aware that SSL/TLS encryption is a compute-intensive operation and will increase the latency of your database connection. SSL/TLS support within Amazon RDS is for encrypting the connection between your application and your DB Instance; it should not be relied on for authenticating the DB Instance itself

Q: Can I encrypt data at rest on my Amazon RDS databases?

Amazon RDS supports encryption at rest for all database engines, using keys you manage using [AWS Key Management Service (KMS)](https://aws.amazon.com/kms/). On a database instance running with Amazon RDS encryption, data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas, and snapshots. Encryption and decryption are handled transparently. For more information about the use of KMS with Amazon RDS, see the [Amazon RDS User's Guide](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html).

You can also add encryption to a previously unencrypted DB instance or DB cluster by creating a DB snapshot and then creating a copy of that snapshot and specifying a KMS encryption key. You can then restore an encrypted DB instance or DB cluster from the encrypted snapshot.

Amazon RDS for Oracle and SQL Server support those engines' Transparent Data Encryption (TDE) technologies

Q: How can I monitor the configuration of my Amazon RDS resources?

You can use [AWS Config](https://aws.amazon.com/config/) to continuously record configurations changes to Amazon RDS DB Instances, DB Subnet Groups, DB Snapshots, DB Security Groups, and Event Subscriptions and receive notification of changes through [Amazon Simple Notification Service (SNS)](https://aws.amazon.com/sns/). You can also create AWS Config Rules to evaluate whether these RDS resources have the desired configurations.

Q: What do “primary” and “standby” mean in the context of a Multi-AZ deployment?

When you run a DB instance as a Multi-AZ deployment, the “primary” serves database writes and reads. In addition, Amazon RDS provisions and maintains a “standby” behind the scenes, which is an up-to-date replica of the primary. The standby is “promoted” in failover scenarios. After failover, the standby becomes the primary and accepts your database operations. You do not interact directly with the standby (e.g. for read operations) at any point prior to promotion. If you are interested in scaling read traffic beyond the capacity constraints of a single DB instance, please see the FAQs on [Read Replicas](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html).

Q: What happens when I convert my RDS instance from Single-AZ to Multi-AZ?

For the RDS for MySQL, MariaDB, PostgreSQL and Oracle database engines, when you elect to convert your RDS instance from Single-AZ to Multi-AZ, the following happens:

* A snapshot of your primary instance is taken
* A new standby instance is created in a different Availability Zone, from the snapshot
* Synchronous replication is configured between primary and standby instances

As such, there should be no downtime incurred when an instance is converted from Single-AZ to Multi-AZ. However, you may see increased latency while the data on the standby is caught up to match to the primary.

Q: What events would cause Amazon RDS to initiate a failover to the standby replica?

Amazon RDS detects and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can resume database operations as quickly as possible without administrative intervention. Amazon RDS automatically performs a failover in the event of any of the following:

* Loss of availability in primary Availability Zone
* Loss of network connectivity to primary
* Compute unit failure on primary
* Storage failure on primary

Note: When operations such as DB instance scaling or system upgrades like OS patching are initiated for Multi-AZ deployments, for enhanced availability, they are applied first on the standby prior to automatic failover. As a result, your availability impact is limited only to the time required for automatic failover to complete. Note that Amazon RDS Multi-AZ deployments do not fail over automatically in response to database operations such as long running queries, deadlocks or database corruption errors.

Q: What happens during Multi-AZ failover and how long does it take?

Failover is automatically handled by Amazon RDS so that you can resume database operations as quickly as possible without administrative intervention. When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary. We encourage you to follow best practices and implement database connection retry at the application layer.

Failovers, as defined by the interval between the detection of the failure on the primary and the resumption of transactions on the standby, typically complete within one to two minutes. Failover time can also be affected by whether large uncommitted transactions must be recovered; the use of adequately large instance types is recommended with Multi-AZ for best results. AWS also recommends the use of Provisioned IOPS with Multi-AZ instances, for fast, predictable, and consistent throughput performance

Q: Can I see which Availability Zone my primary is currently located in?

Yes, you can gain visibility into the location of the current primary by using the [AWS Management Console](https://console.aws.amazon.com/) or DescribeDBInstances API.

Q: After failover, my primary is now located in a different Availability Zone than my other AWS resources (e.g. EC2 instances). Should I be concerned about latency?

Availability Zones are engineered to provide low latency network connectivity to other Availability Zones in the same Region. In addition, you may want to consider architecting your application and other AWS resources with redundancy across multiple Availability Zones so your application will be resilient in the event of an Availability Zone failure. Multi-AZ deployments address this need for the database tier without administration on your part.

Q: When would I want to consider using an Amazon RDS read replica?

There are a variety of scenarios where deploying one or more read replicas for a given source DB instance may make sense. Common reasons for deploying a read replica include:

* Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads. This excess read traffic can be directed to one or more read replicas.
* Serving read traffic while the source DB instance is unavailable. If your source DB Instance cannot take I/O requests (e.g. due to I/O suspension for backups or scheduled maintenance), you can direct read traffic to your read replica(s). For this use case, keep in mind that the data on the read replica may be “stale” since the source DB Instance is unavailable.
* Business reporting or data warehousing scenarios; you may want business reporting queries to run against a read replica, rather than your primary, production DB Instance.
* You may use a read replica for disaster recovery of the source DB instance, either in the same AWS Region or in another Region.

Q: Do I need to enable automatic backups on my DB instance before I can create read replicas?

Yes. Enable automatic backups on your source DB Instance before adding read replicas, by setting the backup retention period to a value other than 0. Backups must remain enabled for read replicas to work.

Q: What is Enhanced Monitoring for RDS?

Enhanced Monitoring for RDS gives you deeper visibility into the health of your RDS instances. Just turn on the “Enhanced Monitoring” option for your RDS DB Instance and set a granularity and Enhanced Monitoring will collect vital operating system metrics and process information, at the defined granularity.

For an even deeper level of diagnostics and visualization of your database load, and a longer data retention period, you can try [Performance Insights](https://aws.amazon.com/rds/performance-insights/).

Q: Which metrics and processes can I monitor in Enhanced Monitoring?

Enhanced Monitoring captures your RDS instance system level metrics such as the CPU, memory, file system and disk I/O among others.

Q: How can I visualize the metrics generated by RDS Enhanced Monitoring in CloudWatch?

The metrics from RDS Enhanced Monitoring are delivered into your CloudWatch Logs account. You can create metrics filters in CloudWatch from CloudWatch Logs and display the graphs on the CloudWatch dashboard.

Q: When should I use CloudWatch instead of the RDS console dashboard?

You should use CloudWatch if you want to view historical data beyond what is available on the RDS console dashboard. You can monitor your RDS instances in CloudWatch to diagnose the health of your entire AWS stack in a single location. Currently, CloudWatch supports granularities of up to 1 minute and the values will be averaged out for granularities less than that.

Q: How can I delete historical data?

Since Enhanced Monitoring delivers JSON payloads into a log in your CloudWatch Logs account, you can control its retention period just like any other CloudWatch Logs stream. The default retention period configured for Enhanced Monitoring in CloudWatch Logs is 30 day